



Patent

Attorney's Docket No. 1032326-000273

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

Jean-Jacques Vandewalle et al.

Application No.: 10/665,905

Filed: September 15, 2003

For: METHOD AND MEANS FOR
MANAGING COMMUNICATIONS
BETWEEN LOCAL AND REMOTE
OBJECTS IN AN OBJECT
ORIENTED CLIENT SERVER
SYSTEM IN WHICH A CLIENT
APPLICATION INVOKES A LOCAL
OBJECT AS A PROXY FOR A
REMOTE ..

) **MAIL STOP AMENDMENT**

) Group Art Unit: 2442

) Examiner: JASON D. RECEK

) Confirmation No.: 7036

AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Office Action dated February 17, 2009, please amend the
above-identified patent application as follows:

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method for managing information exchanges among communicating objects in an object-oriented client server system, said system including first and second object-oriented virtual machines running on counterpart first and second computers in respective server and client roles, and a communication path connection between said computers, said server virtual machine having a run-time environment, the method comprising:

(a) generating a local object at the client machine based upon interface definition of a remote object resident at the server machine, said local object operable-executable as a proxy to ~~a-the~~ remote object ~~resident at the server machine~~; said server machine residing in a smart device; and said client machine having access to the smart device via a smart device reader;

(b) referencing the local object by an application executing at the client machine and causing the local object to marshal parameters;

(c) sending a process level call request by direct method invocation to the run-time environment of the server machine;

(d) responsive to receipt of said request by the server machine's run-time environment, said run-time environment causing the parameters in the request to become unmarshaled, said remote object to be executed, and the results of the execution to be marshaled;

(e) sending a process level return to the client machine as a reply; and

(f) responsive to said reply, unmarshaling the results from said reply by the local object at the client machine.

2. (Previously Presented) The method according to claim 1, wherein plural process call level requests and replies are generated in an alternating manner.

3. (Original) The method according to claim 1, wherein the local object when operating as a proxy at the client machine and the run-time environment when operating at the server machine perform respectively as stubs.

4. (Currently Amended) A method for managing information exchanges between an application executing at a object-oriented virtual machine operable as a client and a remote object resident at another object-oriented virtual machine operable as a server, said server machine having a run-time environment, said client and server having a communication path connection there-between, said communication path connection being operable under a process for originating and sending byte level messages therebetween, comprising:

(a) providing a local object resident at the client machine ~~operable~~ executable as a proxy stub to the remote object resident at the server machine and providing a description of the remote object to enable said run-time environment to also operate as a stub, said server machine residing in a smart device; and said client machine having access to the smart device via a smart device reader; wherein the local object

is generated based upon interface definition of a remote object resident at the server machine;

(b) responsive to a client application call to the local object, marshaling parameters and causing a process level call request to be sent to the run-time environment of the server machine, said sending of the request further including mapping said process level call request into counterpart byte string level messages and transmitting said messages to the server machine;

(c) responsive to receipt of said request messages by the server machine's run-time environment, mapping said messages into a process level call request, unmarshaling the parameters, invoking and executing the remote object, marshaling the results, forming a process level reply, mapping said reply into string byte messages, and transmitting said reply messages to the client machine; and

(d) responsive to the reply messages by the proxy at the client machine, mapping said reply messages into a process level reply, and unmarshaling the results.

5. (Original) The method according to claim 4, wherein said object-oriented virtual machines include Java virtual machines, and further wherein the remote object is an applet, and the local object is an interface description.

6. (Currently Amended) An article of manufacture comprising a machine readable memory having stored therein a plurality of processor executable control program steps for managing information exchanges among communicating objects in an object-oriented client server system, said system including first and second